

Sistem Pendeteksi dan Pelacakan Bola dengan Metode *Hough Circle Transform*, *Blob Detection*, dan *Camshift* Menggunakan AR.Drone

Elki Muhamad Pamungkas^{*1}, Bakhtiar Alldino Ardi Sumbodo², Ika Candradewi³

¹Program Studi Elektronika dan Instrumentasi, FMIPA UGM, Yogyakarta

^{2,3}Departemen Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta

e-mail: ^{*1}elki.muhamad94@gmail.com, ²b.alldino.as@ugm.ac.id, ³ika.candradewi@ugm.ac.id

Abstrak

Parrot AR.Drone merupakan salah satu UAV jenis quadrotor. Quadrotor dapat dioperasikan secara manual dengan remote control dan otomatis memanfaatkan teknologi GPS (Global Positioning System), namun penggunaan GPS dalam misi penjejakan objek memiliki kekurangan yaitu tidak mampu memposisikan quadrotor relatif terhadap objek. Quadrotor memerlukan metode pengendalian lainnya agar dapat melakukan pelacakan objek. Salah satu pendekatannya adalah memanfaatkan pengolahan citra digital. Penelitian ini dirancang sebuah sistem pendeteksian dan pelacakan bola dengan pengolahan citra digital menggunakan pustaka OpenCV yang diimplementasikan pada platform Robot Operating System. Metode yang digunakan adalah *hough circle transform*, *blob detection*, dan *camshift*.

Hasil dari penelitian ini berupa sebuah sistem pada AR.Drone yang mampu mendeteksi dan melakukan tracking bola. Berdasarkan hasil pengujian didapatkan kesimpulan bahwa jarak maksimal sistem mampu mendeteksi bola dengan diameter 20 cm menggunakan metode *hough circle transform* adalah 500 cm sedangkan metode *blob detection* adalah 900 cm. Waktu rata-rata proses pendeteksian dalam mendeteksi bola menggunakan metode *hough circle transform* yaitu 0,0054 detik dan metode *blob detection* yaitu 0,0116 detik. Tingkat keberhasilan tracking bola menggunakan metode *camshift* dari hasil pendeteksian *hough circle transform* adalah 100% sedangkan dari hasil pendeteksian *blob detection* adalah 96,67%.

Kata kunci— Parrot AR.Drone, OpenCV, Robot Operating System

Abstract

Parrot AR.Drone is one type of quadrotor UAV. Quadrotor is operated manually with remote control and automatically using GPS (Global Positioning System), but using GPS in tracking mission an object has disadvantage that can't afford quadrotor position relative to object. Quadrotor require other control methods to perform object tracking. One approach is utilize digital image processing. In this research is designed detection and tracking ball system with digital image processing using OpenCV library and implemented on platform Robot Operating System. The methods which used is *hough transform circle*, *blob detection* and *camshift*.

The results of this research is system on AR.Drone capable of detecting and tracking ball. Based on the test results it was concluded that the maximum distance of system is capable to detecting ball with diameter of 20 cm using *hough transform circle* method is 500 cm and using *blob detection* method is 900 cm. Average time detection process to detect the ball using *hough transform circle* that is 0.0054 second and for *blob detection* method is 0.0116 second. The success rate of tracking the ball using *camshift* method from the results of detection *hough circle transform* is 100% while from result of detection *blob detection* is 96.67%

Keywords— Parrot AR.Drone, OpenCV, Robot Operating System

1. PENDAHULUAN

UAV (*Unmanned Aerial Vehicle*) atau pesawat tanpa awak atau *drone* adalah sebuah mesin yang mampu terbang tanpa membawa pilot atau dikendalikan oleh pilot dari jarak jauh [1]. Beberapa jenis UAV yang ada sekarang antara lain *fixed wing*, *flying-wing*, dan *multirotor*. Penggunaan UAV disesuaikan dengan karakteristik terbang masing-masing. UAV dilengkapi dengan system pengawasan dengan menggunakan kamera dan GPS yang baik untuk pertahanan dan keamanan [2]. **Parrot AR.Drone** merupakan salah satu UAV jenis *quadcopter*. AR Drone 2.0 merupakan salah satu contoh pesawat tanpa awak yang berjenis multirotor dan biasa digunakan untuk riset [3].

UAV dapat dioperasikan secara manual dengan *remote control* dan otomatis memanfaatkan teknologi GPS (*Global Positioning System*). Namun penggunaan GPS masih memiliki kelemahan seperti sulitnya GPS memberikan kendali UAV untuk melakukan pelacakan objek karena sifat GPS yang memiliki orde dalam meter dan objek juga tidak dikenali oleh GPS. UAV memerlukan metode pengendalian lainnya agar dapat melakukan pelacakan objek. Salah satu pendekatannya adalah memanfaatkan pengolahan citra digital. *Computer vision* atau pengolahan citra digital adalah ilmu pemrograman computer untuk memproses dan akhirnya memahami gambar dan video, atau membuat computer dapat melihat [4]. Deteksi objek dalam pengolahan citra digital adalah suatu proses yang digunakan untuk menentukan keberadaan objek tertentu di dalam suatu citra digital [5]. Parrot AR.Drone dapat digunakan untuk melacak dan mengikuti bola bergerak cepat di sepanjang acara olahraga dengan kontrol umpan balik visual menggunakan relatif video berkualitas rendah [6].

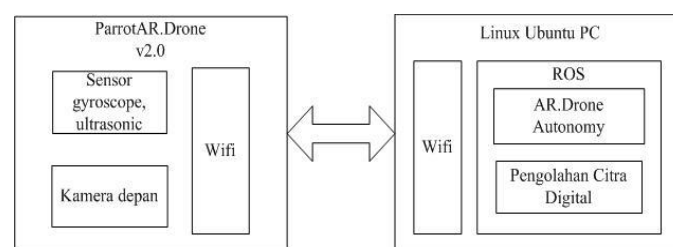
Parrot AR.Drone merupakan salah satu jenis robot yang memiliki kompatibilitas dengan sistem ROS (*Robot Operating System*). ROS adalah *framework* yang berisi *library*, *driver*, maupun peralatan (*tools*) untuk memudahkan pembuatan program yang kompleks pada berbagai *platform* robot. ROS digunakan agar setiap *developer* robot dapat mengembangkan program dengan *style* yang sama, sehingga terjadi kesamaan *programming style* yang bertujuan agar memudahkan komunitas sesama *developer* robot untuk *sharing* dan mengembangkan robot dengan berbagai macam jenis [7].

Oleh karena itu, pengolahan citra digital dapat diterapkan pada AR.Drone dengan menggunakan *platform* ROS. Pengolahan citra digital lebih mengacu pada pengolahan gambar digital melalui komputer [8]. Penelitian yang telah dilakukan ini yaitu mengimplementasikan pengolahan citra digital pada AR.Drone agar mampu melakukan pendeteksian dan *tracking* bola serta menyesuaikan posisi *drone* terhadap bola dengan memanfaatkan pengolahan citra digital atau *computer vision* dan sistem *Robot Operating System* (ROS).

2. METODE PENELITIAN

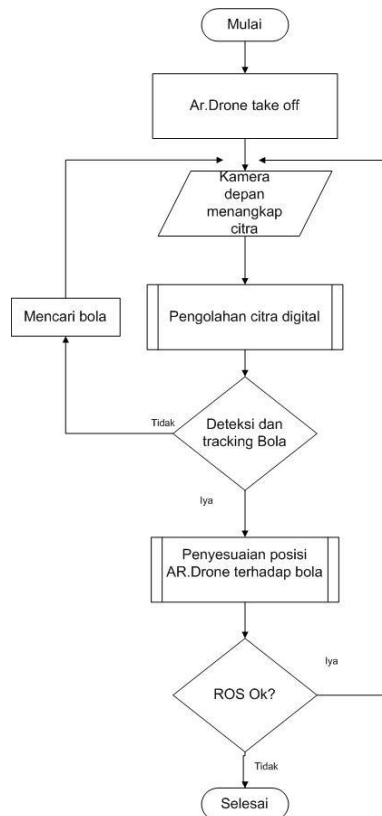
2.1 Analisis Sistem

Sistem yang akan dirancang untuk penelitian ini merupakan suatu sistem pendeteksian dan *tracking* bola berbasis pengolahan citra dengan metode *hough circle transform*, *blob detection* dan *camshift*. Rancangan sistem ditunjukkan pada diagram blok rancangan sistem pada Gambar 1.



Gambar 1 Diagram blok rancangan sistem

Parrot AR.Drone v.2 digunakan untuk mengambil citra menggunakan kamera ketika terbang dan mengirimkan data kamera melalui *wifi* ke laptop. Sedangkan laptop digunakan untuk pemrosesan data dari hasil citra yang ditangkap oleh kamera AR.Drone. Perangkat lunak pengolahan citra dibuat dengan menggunakan pustaka OpenCV. Paket OpenCV diintegrasikan dengan paket AR.Drone untuk mengendalikan AR.Drone. Integrasi paket dilakukan pada *platform* ROS. Pengambilan citra dilakukan secara *realtime* menggunakan kamera AR.Drone. Proses pendeteksian dengan pengolahan citra digital dilakukan menggunakan metode *blob detection* dan *hough circle transform*. Setelah terdeteksi kemudian dilanjutkan dengan *tracking* bola menggunakan metode *camshift*. Hasil dari pengolahan citra digital tersebut dijadikan parameter untuk kendali dari AR.Drone. Gambar 2 menunjukkan diagram alir sistem.

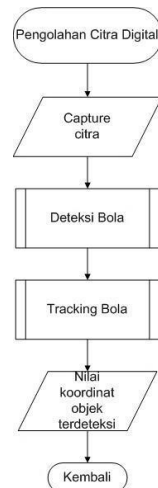


Gambar 2 Diagram alir sistem

2.2 Rancangan Pengolahan Citra Digital

Perancangan pengolahan citra digital penelitian ini meliputi algoritma mendeteksi dan *tracking* objek bola menggunakan software *code editor* pada laptop dengan sistem operasi Linux yang telah ter-install *Robot Operating System* (ROS) serta *library* OpenCV. OpenCV adalah *library open-source*, yang secara luas digunakan dalam aplikasi pengolahan citra untuk mendapatkan pengakuan instan untuk aplikasi tertanam [9].

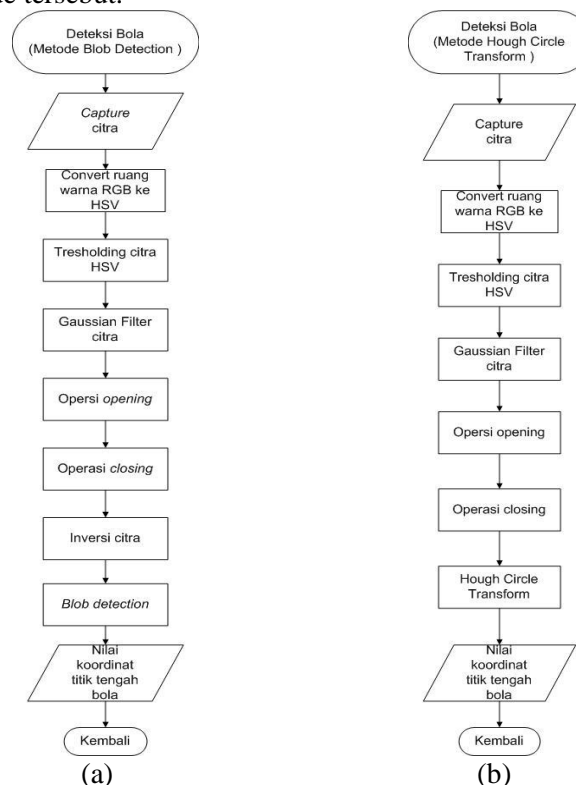
Proses pengolahan citra digital dimulai dengan masukan input citra yang didapat dari kamera depan AR.Drone. Kemudian dilakukan pendeteksian bola dengan menggunakan metode *hough circle transform* atau *blob detection*. Kedua metode pendeteksian diimplementasikan pada program yang terpisah agar dapat diketahui perbandingan kemampuan pendeteksian bola dengan kedua metode tersebut. Selanjutnya adalah proses *tracking* bola yang dilakukan berdasarkan dari hasil deteksi dengan kedua metode pendeteksian. Pada proses pengolahan citra digital pada sistem ini hanya menggunakan satu metode *tracking* yaitu *camshift*. Gambar 3 menunjukkan diagram alir rancangan pengolahan citra digital.



Gambar 3 Diagram alir rancangan pengolahan citra digital

2.2.1 Deteksi bola

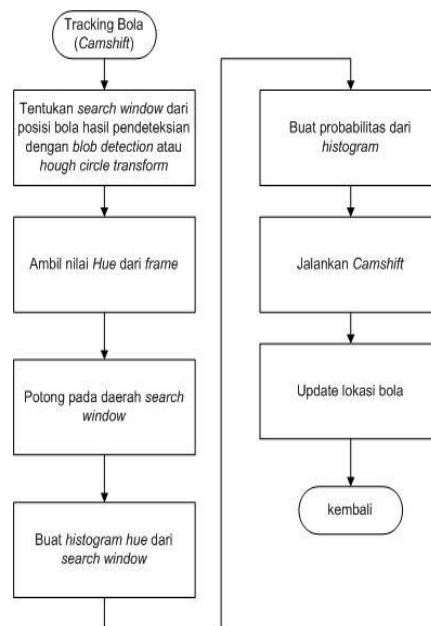
Sebelum proses deteksi bola dilakukan proses *pre processing*. Proses selanjutnya adalah menerapkan metode yang akan diujikan, baik itu *hough circle transform* atau *blob detection*. Pemrosesan citra digital untuk pendeteksian bola menggunakan metode *hough circle transform* atau *blob detection* dilakukan secara terpisah. Pada pendeteksian menggunakan metode *blob detection* dilakukan dengan mengatur parameter blob yaitu parameter ukuran, warna, dan juga bentuk sedangkan pendeteksian menggunakan metode *hough circle transform* dilakukan dengan menentukan nilai *threshold* pendeteksian titik tengah deteksi. Hasil dari proses pendeteksian dengan menggunakan kedua metode pendeteksian tersebut adalah didapatkan nilai koordinat titik tengah objek yang terdeteksi. Gambar 4 menunjukkan diagram alir proses deteksi bola menggunakan dua metode tersebut.



Gambar 4 Diagram alir deteksi bola (a) metode *hough circle transform* dan (b) metode *blob detection*

2.2.2 Tracking bola

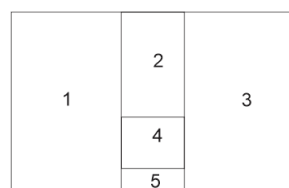
Pada penelitian ini juga menggunakan sistem pengolahan citra digital *tracking* bola. Setelah dilakukan pendeteksian bola dengan menggunakan metode *blob detection* atau metode *hough circle transform* kemudian dilakukan *tracking* bola dengan menggunakan metode *camshift*. *Camshift* merupakan singkatan dari *Continously Adaptive Mean Shift*, yang merupakan pengembangan algoritma *meanshift* yang digunakan secara terus menerus untuk melakukan adaptasi atau penyesuaian terhadap distribusi probabilitas warna yang selalu berubah tiap pergantian *frame* dari *video sequence*[10]. Gambar 5 menunjukkan diagram alir *tracking* bola.



Gambar 5 Diagram alir *tracking* bola

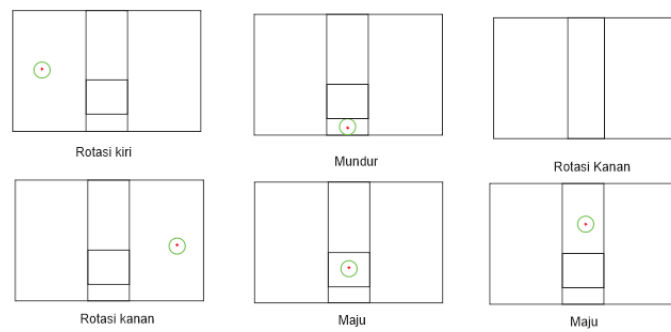
2.3 Rancangan Penyesuaian Posisi AR.Drone Terhadap Bola

Setelah pendeteksian dan *tracking* bola dapat dilakukan, maka proses selanjutnya adalah penyesuaian posisi AR.Drone terhadap bola. Untuk itu *frame* dilakukan pembagian atau segmentasi menjadi lima bagian yaitu *frame* bagian 1, *frame* bagian 2, *frame* bagian 3, *frame* bagian 4, dan *frame* bagian 5. Gambar 6 menunjukkan hasil segmentasi dari *frame*.



Gambar 6 Hasil segmentasi *frame*

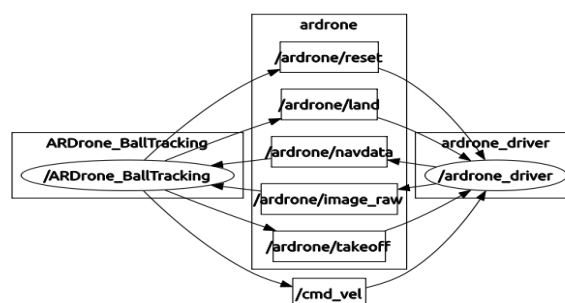
AR.Drone akan menyesuaikan posisinya agar bola yang telah di-*tracking* dan dideteksi berada pada *frame* bagian 4 dengan melakukan rotasi kiri, rotasi kanan, gerak maju dan gerak mundur. Perintah penyesuaian posisi AR.Drone terhadap bola didasarkan pada posisi objek bola yang mampu di-*tracking* dan dideteksi pada *frame*. Gambar 7 menunjukkan Pergerakan AR.Drone berdasarkan posisi bola.



Gambar 7 Pergerakan AR.Drone berdasarkan posisi bola

2.4 Rancangan Robot Operating System

Pada penelitian ini menggunakan *Robot Operating System* versi keempat yaitu ROS Indigo. Pustaka ROS digunakan sebagai program pengendali Parrot AR.Drone yang ditanam pada *platform* Ubuntu 14.04. Prosedur penggunaan ROS diantaranya yaitu pustaka ROS harus ter-install terlebih dahulu pada PC kemudian dibuat sebuah *workspace* yang berisi program pengolahan citra digital untuk pendeteksian, *tracking* bola dan program kendali gerak AR.Drone. Pada ROS terdapat *node*, *publisher*, *subscriber* dan *topic*. *Node* pada ROS menyerupai sebuah aplikasi yang berjalan diatas sebuah sistem operasi. *Publisher* digunakan untuk mengirimkan data ke sebuah *topic*. *Subscriber* digunakan untuk menerima data yang terdapat pada sebuah *topic*. Sedangkan *topic* adalah sebuah tempat dimana *node* dapat mengirimkan atau menerima data atau *message*. Gambar 8 menunjukkan hubungan antar *node* pada ROS.



Gambar 8 Hubungan antar *node*

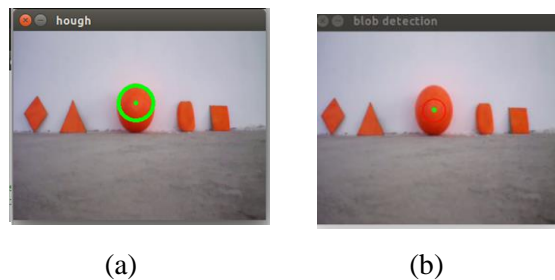
Terdapat dua buah *node* yang digunakan pada sistem pendeteksian dan *tracking* bola ini yaitu *node* ARDrone_BallTracking dan *node* ardrone_driver. *Node* ARDrone_BallTracking merupakan *node* yang digunakan untuk semua pemrosesan data dari sistem. Pada *node* ARDrone_BallTracking terdapat dua buah bagian utama yaitu pengolahan citra digital dan pengendalian AR.Drone. *Node* ARDrone_BallTracking akan dikomunikasikan dengan *node* /ardrone_driver untuk dapat menerima dan mengirimkan data yang dibutuhkan oleh sistem. *Node* /ardrone_driver adalah *driver* yang terdapat pada AR.Drone. Terdapat beberapa *topic* pada *node* /ardrone_driver diantaranya yaitu /ardrone/reset, /ardrone/land, /ardrone/navdata, /ardrone/image_raw, /ardrone/takeoff dan /cmd_vel.

3. HASIL DAN PEMBAHASAN

Pada bab ini akan dibahas mengenai hasil dari pengujian sistem serta pembahasannya. Secara garis besar dibagi menjadi tiga bagian yaitu pengujian kemampuan sistem dalam melakukan pendeteksian dan *tracking* bola serta pengujian keberhasilan sistem dalam melakukan penyesuaian posisi AR.Drone terhadap bola.

3.1 Pengujian pendeteksian bola

Pengujian pendeteksian deteksi bola bertujuan untuk menyeleksi objek yang bukan bola. Pendeteksian menggunakan metode *blob detection* dilakukan dengan mengatur parameter blob yang akan dideteksi. Parameter yang diatur adalah parameter *minimal circularity*. Parameter *circularity* maksudnya adalah kecenderungan sebuah blob yang diinginkan dengan sebuah lingkaran. Pendeteksian bola dengan menggunakan kamera bagian depan AR.Drone diberikan nilai *minimal circularity* 0,82. Setelah dilakukan pengujian dengan menyandingkan bersama objek selain bola, sistem berhasil melakukan pendeteksian dan hanya objek bola yang dapat dideteksi. Gambar 9 menunjukkan keberhasilan pendeteksian dengan metode *hough circle transform* dan metode *blob detection*.



Gambar 9 Deteksi bola (a) metode *hough circle transform* dan (b)metode *blob detection*

3.2 Pengujian kinerja sistem pendeteksian terhadap jarak AR.Drone dengan bola

Pengujian ini dilakukan untuk mengetahui kinerja dari sistem pendeteksian menggunakan metode *hough circle transform* dan *blob detection* dengan melakukan variasi jarak antara AR.Drone dengan bola. Variasi jarak bola dengan AR.Drone yang digunakan adalah 100 cm, 200 cm, 300 cm, 400 cm, 500 cm, 600 cm, 700 cm, 800 cm dan 900 cm. Penyesuaian jarak antara AR.Drone dengan bola dilakukan dengan alat ukur meteran. Data yang diambil dari pengujian ini adalah sejumlah 1000 *frame*. Data kinerja sistem pendeteksian menggunakan metode *hough circle transform* dan *blob detection* ditunjukkan pada Tabel 1.

Tabel 1 Data kinerja sistem pendeteksian dengan menggunakan metode *hough circle transform* dan *blob detection*

| Jarak AR.Drone ke bola | Tingkat Keberhasilan Pendeteksian (dalam persen) | |
|------------------------|--|-----------------------|
| | <i>Hough circle transform</i> | <i>Blob detection</i> |
| 100 cm | 100 % | 100 % |
| 200 cm | 100 % | 100 % |
| 300 cm | 100 % | 100 % |
| 400 cm | 95,50 % | 99,90 % |
| 500 cm | 44,34 % | 100 % |
| 600 cm | 0 % | 100 % |
| 700 cm | 0 % | 99,80 % |
| 800 cm | 0 % | 98,70 % |
| 900 cm | 0 % | 84,88 % |
| Rata-rata | 48,87 % | 98,14 % |

Mengacu pada Tabel 1, kinerja deteksi bola dengan dua metode pendeteksian pada keadaan statis memiliki tingkat keberhasilan yang jauh berbeda. Pada pendeteksian menggunakan metode *hough circle transform* rata-rata keberhasilannya adalah sebesar 48,87 % sedangkan keberhasilan pendeteksian rata-rata dengan menggunakan metode *blob detection* adalah sebesar 98,14 %.

3.3 Pengujian waktu proses pendeteksian bola

Pengujian waktu proses pendeteksian bola dilakukan untuk mengetahui perbandingan waktu pemrosesan pendeteksian. Data yang diambil adalah sebanyak 1000 data, selanjutnya dilakukan perhitungan rata-rata untuk diketahui selisih waktu proses pendeteksian dengan menggunakan metode *hough circle transform* dan metode *blob detection*. Pengujian ini dilakukan pada *node* ARdrone_BallTracking yaitu perangkat PC dengan spesifikasi *processor* Intel Core i3-2350M dengan kecepatan 2.30 GHz dan RAM sebesar 6GB. Tabel 2 menunjukkan sebagian data dari waktu proses pendeteksian bola.

Tabel 2 Data waktu proses pendeteksian bola

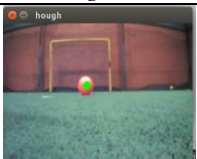


| No | Waktu Pendeteksian (detik) | |
|------------------|-------------------------------|-----------------------|
| | <i>Hough circle transform</i> | <i>Blob detection</i> |
| 1 | 0.0459 | 0.0481 |
| 2 | 0.0045 | 0.0092 |
| . | . | . |
| . | . | . |
| 1000 | 0.0084 | 0.0086 |
| Rata-rata | 0.0054 | 0.0116 |

Pada Tabel 2 Disajikan data waktu proses pendeteksian bola dengan menggunakan metode *hough circle transform* dan metode *blob detection*. Waktu rata-rata yang dibutuhkan untuk melakukan pendeteksian menggunakan metode *hough circle transform* adalah 0.0054 detik dengan nilai standar deviasi sebesar 0.001978731 sedangkan waktu rata-rata yang dibutuhkan menggunakan metode *blob detection* adalah 0.0116 detik dengan nilai standar deviasi sebesar 0.003493475.

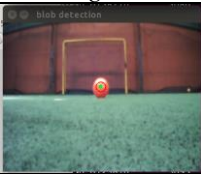
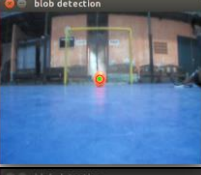

3.4 Pengujian pendeteksian bola pada beberapa background

Pengujian pendeteksian pada beberapa kondisi *background* ini dilakukan agar dapat melihat kemampuan sistem dalam melakukan pendeteksian dengan variasi kondisi *background*. Terdapat 3 variasi kondisi *background* yang digunakan. Data yang diambil adalah sebanyak 1000 *frame*. Tabel 3 dan Tabel 4 menunjukkan hasil pendeteksian bola pada variasi kondisi *background* dengan *hough circle transform* dan *blob detection*

Tabel 3 Data hasil deteksi bola pada variasi kondisi *background* dengan *hough circle transform*

| No | Nilai <i>threshold</i> | <i>Background</i> | Tingkat keberhasilan |
|-----------|--|---|----------------------|
| 1 | Hue : 0-100 Saturation: 85-255 Value : 180-255 |  | 100 % |
| 2 | Hue : 0-100 Saturation: 85-255 Value : 180-255 |  | 96,00% |
| 3 | Hue : 0-100 Saturation: 85-255 Value : 180-255 |  | 100 % |
| Rata-rata | | | 98,67 % |

Tabel 4 Data hasil deteksi bola pada variasi kondisi *background* menggunakan *blob detection*

| No | Nilai <i>threshold</i> | <i>Background</i> | Tingkat keberhasilan |
|-----------|--|---|----------------------|
| 1 | Hue : 0-100 Saturation: 85-255 Value : 180-255 |  | 96,80 % |
| 2 | Hue : 0-100 Saturation: 85-255 Value : 180-255 |  | 99,30 % |
| 3 | Hue : 0-100 Saturation: 85-255 Value : 180-255 |  | 100 % |
| Rata-rata | | | 98,70 % |

Berdasarkan pada data yang disajikan pada Tabel 3 dan Tabel 4 dapat dilihat bahwa pendeteksian bola mampu dilakukan dengan baik. Pendeteksian bola nilai *threshold* HSV yang sama yaitu *hue* 0-100, *saturation* 85-255 dan *value* 180-255 pada ketiga variasi *background* memiliki kinerja yang baik. Pendeteksian bola menggunakan metode *hough circle transform* memiliki tingkat keberhasilan pendeteksian rata-rata sebesar 98,67% sedangkan pendeteksian menggunakan metode *blob detection* memiliki tingkat keberhasilan rata-rata sebesar 98,70%.

3.5 Pengujian Tracking Bola dengan Metode Camshift

Pengujian *tracking* bola dilakukan untuk mengetahui kemampuan sistem dalam melakukan *tracking* bola dengan metode *camshift* terhadap dua hasil deteksi bola yaitu dengan metode *hough circle transform* dan metode *blob detection*. Data diambil sebanyak 20 *frame* percobaan yang dilakukan pada satu sampel yang dilakukan secara kontinu. Untuk mendapatkan nilai akurasi dari *tracking* digunakan *confusion matrix*. Pada pengujian *tracking* ini diberikan dua buah objek bukan bola dengan warna oranye yang berada pada satu *frame* pengujian *tracking*.

Sebuah *confusion matrix* menampilkan jumlah prediksi yang benar dan tidak benar yang dibuat oleh model dibandingkan dengan klasifikasi yang sebenarnya dalam data uji. Tabel 5 menunjukkan *confusion matrix* hasil pengujian *tracking* bola dari hasil deteksi menggunakan metode *hough circle transform*. Selanjutnya, Tabel 6 menunjukkan *confusion matrix* hasil pengujian *tracking* bola dari hasil deteksi menggunakan metode *blob detection*. Matriks $n \times n$, dimana n adalah jumlah kelas. Masukan dari *confusion matrix* memiliki antara lain:

True Positive = jumlah objek yang benar merupakan sebuah bola.

False Positive = Hasil yang terpenuhi, ketika sebenarnya tidak terpenuhi.

False Negative = ketika mengindikasikan gagal, namun seharusnya berhasil.

True Negative = jumlah objek yang tidak boleh terdeteksi dan benar tidak.

Akurasi (AC) adalah proporsi jumlah prediksi yang benar. Untuk menghitung keakuratannya digunakan rumus seperti pada persamaan (1) sebagai berikut:

$$\text{Akurasi} = \frac{\text{true positive} + \text{true negative}}{\text{true positive} + \text{false positive} + \text{false negative} + \text{true negative}} \times 100\% \quad (1)$$

Tabel 5 Confusion matrix hasil pengujian *tracking* bola berdasarkan hasil deteksi *hough circle transform*

| | | Sistem | |
|--------|---------|---------|---------|
| | | Negatif | Positif |
| Aktual | Negatif | 40 | 0 |
| | Positif | 0 | 20 |

Maka nilai akurasi dari pengujian ini adalah :

$$\text{Akurasi} = \frac{20+40}{20+0+0+40} \times 100\% \\ = 100\%$$

Tabel 6 Confusion matrix hasil pengujian *tracking* bola berdasarkan hasil deteksi *blob detection*

| | | Sistem | |
|--------|---------|---------|---------|
| | | Negatif | Positif |
| Aktual | Negatif | 40 | 0 |
| | Positif | 2 | 18 |

Maka nilai akurasi dari pengujian ini adalah :

$$\text{Akurasi} = \frac{18+40}{18+0+2+40} \times 100\% \\ = 96,67\%$$

Pada Tabel 5 dapat terlihat bahwa *tracking* bola dengan *camshift* menggunakan metode *hough circle transform* memiliki kecenderungan dapat terlacak. Pada dua puluh *frame* percobaan yang dilakukan pada satu sampel yang dilakukan secara kontinu, terdapat dua puluh *frame* bola yang terlacak sebagai sebuah bola dan dari dua puluh *frame* percobaan juga terdapat empat puluh objek yang bukan sebuah bola sehingga tingkat akurasi *tracking*-nya yaitu 100%.

Hasil pengujian yang berbeda pada *tracking* bola menggunakan hasil deteksi *blob detection* ditunjukkan pada Tabel 6, dapat terlihat bahwa terdapat delapan belas *frame* bola yang terlacak sebagai sebuah bola dan dari dua puluh *frame* percobaan juga terdapat empat puluh objek yang bukan sebuah bola. Hasil *confusion matrix* menunjukkan bahwa tingkat akurasi dari *tracking* bola dengan metode *camshift* dari hasil deteksi *blob detection* adalah 96,67%. Gambar 10 menunjukkan hasil dari *tracking* bola dengan metode *camshift*.

Gambar 10 Hasil *tracking* bola dengan metode *camshift*

3.6 Pengujian Penyesuaian Posisi AR.Drone terhadap Bola

Pengujian penyesuaian posisi AR.Drone terhadap bola dilakukan agar dapat mengetahui tingkat keberhasilan sistem dalam menyesuaikan posisi AR.Drone terhadap bola. Pengujian penyesuaian posisi AR.drone terhadap bola ini dilakukan dengan menerbangkan AR.drone pada jarak 300 cm terhadap bola. AR.Drone diterbangkan dengan menggunakan ketinggian *default* dari AR.Drone yaitu 75 cm.

Pengujian ini dilakukan sebanyak 20 kali yaitu dilakukan 10 kali pengujian pada setiap metode pendeteksian. Pengujian dikatakan berhasil apabila AR.Drone berhasil mendeteksi dan *tracking* bola serta dapat melakukan respon gerakan sehingga bola yang terdeteksi dan dapat di-*tracking* berada tepat di depan AR.Drone. Pengujian dikatakan tidak berhasil apabila AR.Drone tidak berhasil menyesuaikan posisi walaupun sudah dapat melakukan pendeteksian dan *tracking* bola. Data navigasi direkam dan dapat ditampilkan pada rosbag.

Penyesuaian AR.Drone berdasarkan hasil deteksi *hough circle transform* adalah berhasil melakukan penyesuaian posisi sebanyak 6 kali dari total 10 kali percobaan sehingga tingkat keberhasilannya adalah 60%. Sedangkan penyesuaian posisi AR.Drone berdasarkan hasil deteksi *blob* adalah berhasil melakukan penyesuaian posisi sebanyak 7 kali dari total 10 kali percobaan sehingga tingkat keberhasilannya adalah 70%. Namun, secara garis besar waktu keberhasilan misi penyesuaian posisi AR.Drone terhadap bola dengan menggunakan hasil deteksi dari *hough circle transform* lebih cepat dibandingkan dengan menggunakan hasil deteksi dari *blob detection*.

4. KESIMPULAN

Berdasarkan penelitian yang dilakukan telah berhasil dibuat sistem pendeteksian dan *tracking* bola dengan metode *hough circle transform*, *blob detection*, dan *camshift* menggunakan media AR.Drone. Berdasarkan data yang didapat dan pembahasan yang dilakukan dapat diambil kesimpulan sebagai berikut:

1. Jarak maksimal sistem mampu mendeteksi bola dengan diameter 20 cm menggunakan metode *hough circle transform* adalah 500 cm sedangkan metode *blob detection* adalah 900 cm.
2. Waktu rata-rata proses pendeteksian dalam mendeteksi bola pada PC dengan spesifikasi *processor* Intel Core i3-2350M dengan kecepatan 2.30 GHz dan RAM sebesar 6GB menggunakan metode *hough circle transform* adalah 0.0054 detik dengan nilai standar deviasi sebesar 0.001978731 dan metode *blob detection* adalah 0.0116 detik dengan nilai standar deviasi sebesar 0.003493475.
3. Sistem pendeteksian dengan menggunakan metode *blob detection* dan metode *hough circle transform* memiliki kehandalan yang baik dalam mendeteksi terhadap beberapa variasi *background* dengan nilai *threshold* HSV untuk bola warna oranye yaitu *hue* 0-100, *saturation* 85-255, dan *value* 180-255.
4. Tingkat keberhasilan *tracking* bola menggunakan metode *camshift* berdasarkan dari hasil pendeteksian *hough circle transform* adalah 100% sedangkan berdasarkan dari hasil pendeteksian *blob detection* adalah 96,67%.

5. SARAN

Berikut saran-saran yang dapat dilakukan untuk pengembangan penelitian selanjutnya yang sejenis.

1. Penggunaan sistem kendali pada AR.Drone dengan menambahkan kendali PID pada sistem sehingga dapat menjaga kestabilan dalam melakukan terbang dan misi.
2. Metode pengolahan citra digital dapat dikembangkan dengan menggunakan metode pengenalan pola atau dengan metode ekstraksi fitur agar objek yang dideteksi tidak hanya menggunakan bentuk dan warna

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada semua pihak yang telah membantu dalam mengerjakan penelitian ini.

DAFTAR PUSTAKA

- [1] H.Chao, Y. Cao, and Y.Chen, 1995, Autopilots for Small Unmanned Aerial Vehicles: A Survey, *International Journal of Control, Automation, and System*, 1995.
- [2] S.D. Handford, 2005, A Small Semi-Autonomous Rotary-Wing Unmanned Air Vehicle, The Pennsylvania State University, University Park, PA.
- [3] Ritya L., Andriana G., Indah G., 2005, Pengendalian AR Drone 2.0 dan Pengambilan Data Citra Berdasarkan Koordinat GPS, Universitas Telkom, Bandung.
- [4] Culjak, I., Abram, D., Pribanic, T., Dzapo, H. dan Cifrek, M., 2012, A brief introduction to OpenCV, *MIPRO, 2012 Proceedings of the 35th International Convention*, Croatia, May 21-25.
- [5] Nagataries, D., Hardiristanto, S., Purnomo, M.H., 2012, Deteksi Objek pada Citra Digital Menggunakan Algoritma Genetika untuk Studi Kasus Sel Sabit.
- [7] Ataka A., 2014, ROS, Satu Langkah Menuju Pemrograman Robot yang Lebih Mudah, <http://ieeesb.ft.ugm.ac.id/ros-satu-langkah-menuju-pemrograman-robot-yang-lebih-mudah/>, diakses pada 11 Ferbuari 2015.
- [8] Weickert, J., 2001, *Image Processing and Computer Vision*. [Online]. Available from: <http://books.google.com/books?id=eSu5I9pU3rUC&pgis=1>.
- [9] Deepthi, R. dan Sankaraiah, S., 2011, Implementation of mobile platform using Qt and OpenCV for image processing applications, *Open Systems (ICOS), 2011 IEEE* [Online] 284–289. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6079235
- [10] Bradski, G., dan Kaehler, A. (2008). *Learning OpenCV: Computer Vision with the OpenCV=Library*. OReilly Media Inc (Vol. 1) .doi:10.1109/MRA.2009.933612